

## Question Bank - OOPS.

BCS10T1006

- Ques 1. Differentiate between data abstraction and data encapsulation.

Abstraction

Encapsulation

- |  |  |
|--|--|
| 1. Abstraction is the process or method of gaining the information.      | Encapsulation is the process or method to contain the information. |
| 2. In abstraction, Problems are solved at the design or interface level. | In encapsulation problems are solved at the implementation level.  |
| 3. Abstraction is the method of hiding the unwanted information.         | Encapsulation is a method to hide the data in a single entity.     |
| 4. We can implement abstraction using abstract class and interfaces.     | Encapsulation can be implemented using access modifier.            |

Q1  
Page 3

Q2. Discuss the use of Public, Private and Protected access specifiers, and their visibility in the class.

Ans → Public ⇒ members are accessible from outside the Class.

Private ⇒ members cannot be accessed from outside the class.

Protected ⇒ members can not be accessed from outside the class. However, they can be accessed in inherited classes.

The Protected access specifier allows the class the member belongs to, friends and derived classes to access the member.

Q3 Discuss default constructor and Parameterized constructor with the help of an example in C++.

Ans → Default Constructor  
i) A constructor that is automatically generated by the compiler in the absence of any programmer defined constructor.

Parameterized

i) A constructor that is created by the programmer with one or more parameters to initialize the instance variables of a class.

- ii) Has no parameters
- iii) If the programmer hasn't written a constructor, the default constructor is automatically called.
- ii) Has one or more parameters
- iii) Programmer should write his own parameterized constructor.

Parameterized Constructor  $\Rightarrow$  A constructor that accepts or receive parameters is called Parameterized constructor.

Syntax:- Class-name (Para1, Para2...).  
{  
    || code.  
}

Q-4 Write down the use of destructor in C++.

Ans:-

- i) Destructor works opposite to constructor.
- ii) It destruct the objects of classes.
- iii) It can be defined only once in a class.
- iv) It is used to remove an object.

Q-5 What is the need of constructor? How it is different from the member function?

Ans:- A constructor is a special method of a class that initializes new objects.

Without a constructor you can't create instances of the class.

Constructor

Member function.

Constructor doesn't have return type.

Member function has a return type.

Q⇒ 6. What is a Static data member? How they are used in Static function? Explain with Suitable illustrations.

Ans⇒ It is a variable which is declared with the static keyword.

ii) It is also known as Class member.

iii) It is always initialized with zero because its default value is zero.

iv) It retains its values.

\* By declaring a function member as static you make it independent of any particular object of the class.

Static member function are special functions used to access the static data members or other static member functions.

Q⇒ 7. Define Class and Object.

Ans⇒ Class is a user defined data type. It is a collection of data members and member functions which can accessed and used by creating object of that class.

Object⇒ An Object is an instance of a class. Object have states and behaviours.

Ex  $\Rightarrow$  A dog has States - colour, name, barked as well as behaviours.

Object is created with a class name in C++.

Q  $\Rightarrow$  8. What do you mean by dynamic binding? How it is useful in OOP?

Ans  $\Rightarrow$  Dynamic binding also known as late-binding.  
When type of object is determined at run-time using it is known dynamic binding.

It provides a mechanism for selecting the function to be executed from various function alternatives at the run-time.

Q  $\Rightarrow$  9. Explain the use of friend function with the help of suitable example.

Ans  $\Rightarrow$  A friend function in C++ is defined as a function that can access private, protected and public members of a class.

The friend function is declared using the friend keyword inside the body of class.

The use of friend function to access the private or protected members of a class.

Ex → A simple example of a friend class.

```
#include <iostream>
```

Using namespace std,  
Class A

```
{ int x=4;  
    friend Class B; // Friend Class
```

}, Class B

```
{ Public:  
    void display(A&a);
```

```
    cout << "value of x is: " << a.x;
```

}, int main()

```
{ A a;  
    B b;  
    b.display(a);  
    return 0;
```

Output: value of x is 4.

Ques 1. What is the need of overloading operators and functions?

Ans  $\Rightarrow$  Operator overloading means providing multiple definition for the same operator.

Operator Overloading

A feature in C++ that enables the redefinition of operators. This feature operates on user defined objects.

Function Overloading  $\Rightarrow$  function overloading reduces the invention of different function names and used to perform similar functionality by more than one function.

Ques 11 How do we invoke Constructor? Can we have more than one constructor in a class? If yes, explain the need for such a situation.

Ans  $\Rightarrow$  You can call another constructor via the this (...) keyword. (When you need to call a constructor from the same class).

Yes we have more than one constructor in a class. If an instance constructor is defined without parameters, that constructor becomes the default instance constructor of the class.

Ques 12 Write down the example to overload unary and binary operators in C++.

Ans ⇒

Examples of unary operators are.

The increment operator `++` and decrement

operator `--` are examples of unary operator.

It is used to perform operation on user defined data type.

Examples of binary operators are

- i) Multiplication of two operands (`*`)
- ii) Division of two Operands (`/`)
- iii) Modulus Operator - The result is the remainder of division (`%`)

Unary ⇒ e.g ⇒ Assume that Class Distance takes two member object i.e. feet and inches. Create a function by which Distance object should decrement the value of feet and inches by 1.

Binary ⇒ e.g ⇒ Assume that Class Distance takes two member i.e feet and inches. Create a function by which Distance object should add the value of feet and inches by 1.

Q→13 State the use of Scope resolution operator in C++.

Ans → The `::` (Scope resolution) Operator is used to get hidden names due to variable scopes so that you can still use them.

The Scope resolution operator can be used as both unary and binary.

ii) It is use to define a function outside a Class.

- iii) To access a Class's Static variables.
- iv) In case of multiple inheritance.
- v) for namespace.

Q→14 Compare and contrast the Structured Programming and Object oriented Programming.

Ans → Structured programming

Object oriented programming

- |   |  |
|---|--|
| i) It gives more importance of code.                        | ii) It gives more importance to data.                      |
| ii) It generally follows <sup>bb</sup> Top Down Approach.   | ii) It generally follows <sup>bb</sup> Bottom-up Approach. |
| iii) Programs are divided into small programs or functions. | iii) Programs are divided into objects or entities.        |

Q>15 What is dynamic constructor? Explain with suitable example.

**Ans:-** Dynamic Constructor is used to allocate the memory to the objects at the run time. By using this we can dynamically initialize the objects.

Ex:- #include <iostream>  
using namespace std;

Class geeks {  
    const char \*p;

public:

    geeks ()  
{

    p = new char [6];

    p = "geeks";

    Void display ()

{

    cout << p << endl;

}

int main ()

{

    geeks obj;

    obj.display ();

}

Q→ 17. Compare and contrast late binding and early binding.

Late binding

Early binding

Early binding

Late binding

- i) Early Binding happens at compile time. Late binding happens at run time.
- ii) Early binding uses the class information to resolve method calling. Late Binding uses the object to resolve method calling.
- iii) Early binding is also known as static binding. iii) Late binding is also known as dynamic binding.
- iv) Execution Speed is faster in early binding. Execution Speed is lower in late binding.

Q→ 18. In what do you mean by implicit and explicit call of constructor? Explain with example.

Ans→ Explicit means done by the programmer.

Implicit means done by the JVM or Tool.

Ex:- int a=42; Implicit  
float b=a;

b will contain typecast value of a, because while assigning value to b compiler typecast the value of a into float then assigns it to b.

## Explicit

```
ex - float a=42.12;  
      int b=(int)a;
```

Here we explicitly converted float value of a to int while assigning it to int b. (int) is the type casting operator with the type in which you want to convert.